Robotic mapping assisted by local magnetic field anomalies

Haiyang Zhang, Fred Martin
Computer Science Department
University of Massachusetts Lowell
Lowell, MA 01854, USA
hzhang@cs.uml.edu, fredm@cs.uml.edu

***Abstract* — This paper presents a method to incorporate measurement of local magnetic field anomalies into the SLAM (Simultaneous Localization And Mapping) algorithm. One of the key problems of SLAM is loop closure, which means to map the same place into the same location on the generated map when the place is revisited by the robot. It is particularly important for large area map consistency. Steel structures, furniture and equipment inside a building disturb the natural magnetic field of the earth. These local anomalies of magnetic field are usually considered as noise when using an electronic compass on a robot. But, in this paper, we utilize these data to help solving the SLAM loop closure problem, since the magnetic anomaly patterns are different across a given building and stay relatively stable over time. We used particle filter as the basic algorithm for SLAM. In the weight calculation step of the particle filter algorithm, it assigns a weight value to each particle based on its matching score to the current sensor data. This step allows us to incorporate different sensor data from both laser range finder, and magnetometer into SLAM process. To verify the effectiveness of this method, we have done experiments involving a robot runs through a building hallway. At some point, it is hijacked to an unknown place in the same hallway, and continues to run. The SLAM assisted by local magnetic field anomaly data has generated more consistent maps than those without the assistance.**

Keywords – robotic mapping; localization; SLAM; magnetic field; anomaly

I. INTRODUCTION

SLAM (Simultaneous Localization And Mapping) algorithm [1, 2] generates a map while a robot travels through an unknown space, and determines the current location simultaneously. The inputs to the SLAM algorithm are usually the odometry (encoder) and laser range data. Based on the historical data and current observations, the robot's most likely location is determined by a statistical model, such as Extended Kalman Filters, or Particle Filters. The map can be feature (landmark) based or occupancy grid based.

Since the introduction of SLAM algorithm, recent research has been done on alternative methods to sense the robot location and new algorithms for loop closure. Loop closure means to map the same place into the same location on the generated map when the place is revisited by the robot. It is particularly important for large area map consistency. For example [3], machine learning is used to achieve loop closure based on laser data. The learning algorithm, a binary classifier based on boosting, first gets trained on existing data pairs. The training is based on features that are rotational invariants, such as average range, number of clusters, etc. Then, in an unknown environment, it compares the current laser scan with historical data, and find out matches. The acquired data are processed by SLAM based on ESDF (Exactly Sparse Delayed-state Filter) to generate the global map. Another example [4] is to use active-SLAM for loop closure. Active SLAM algorithm actively controls the robot's action based on just-generated map to improve localization and mapping accuracy. The algorithm is designed to explore unknown area with loops on the path. There are two exploration modes – regular exploration and loop-closing exploration modes. In regular mode, the robot explore the unknown areas with preference of directions leading towards faster loop closing - explore smaller loops first. When a possible loop closure (reaching a previous location) is detected, the robot enters loop-closing exploration mode. It takes a series of sensing actions to validate the loop closure and optimize the existing map. Particle filter is used in loop closing validation. Their experiments show that the strategy, exploring the smaller loop first, works better. Reference [7] presented DP-SLAM (Distributed Particle SLAM), which is an optimized SLAM algorithm. It based on laser data only, and works without assumptions on landmarks. The enhanced precision enables it to close loop in longer range.

In additional to laser range data based algorithm, alternative environmental signals are explored for robotic mapping. For example WiFi-SLAM [5] is a SLAM algorithm based on WiFi signal strength. Inside or around a building, we usually can receive multiple WiFi signals no matter where you go. The signal strength gradually becomes weaker as the distance from the wireless access point becomes longer. By monitoring signal strength from multiple access points, we can figure out the current location. They use a statistical model, Gaussian process latent variable model (GP-LVM), to analyze the WiFi signal data. There are multiple WiFi signals, which correspond to multiple dimensional data. And the unknown location coordinate (x,y) is treated as latent variable. The GP-LVM model maps the signal strengths to location. Another example of environmental signals is the magnetic field. The paper [6] described a localization technique based on the local anomalies of the ambient magnetic field. In SLAM process, we may use electronic compass to measure the robot pose (orientation). In comparison to encoder based pose estimation, electronic compass doesn't accumulate error over time. But the magnetic field inside a building is affected by the surrounding structures like steel frame, furniture, and electronic devices. These local magnetic field anomalies affect electronic compass readings. Local magnetic field anomalies are usually considered to be noise. But the paper [6] found the magnetic field anomalies are relatively stable and can be used for localization. Monte Carlo Localization is used to estimate the location. The study included experiments of 1D localization in building hallways without simultaneously generating 2D maps.

In this paper, we present a method to incorporate local magnetic field anomaly data into the SLAM algorithm to assist loop closure in 2D simultaneous localization and mapping. We first verified in our building that the magnetic field changes significantly when moving to different places of the hallway, but the anomalies are fairly stable over time. Then, we incorporate the magnetic data into particle filter based SLAM algorithm to help localize the hijacked robot on the map generated previously. Our experiment results show the magnetic anomaly data helped the robot localize itself on the previously visited location on the map (loop closure), and generated consistent map after hijacking.

II. Algorithm and Method

The Particle Filter [2] is used as the basic algorithm for localization and loop closure during mapping process. The idea is to use a set of particles to represent the probabilistic distribution of possible robot states. Each particle contains one possible state of the robot. For our robot, the state X includes the location (x, y), and pose (yaw). A state with higher probability is represented by more particles containing this state.

A. The basic steps of the particle filter algorithm

1) Set the initial distribution of the particles

In this case, we don’t have any knowledge of the robot initial state, so just initialize all particles to x=0, y=0, yaw=0. They will be updated in the following steps based on control and sensor data.

2) Apply control data

Read a new set of control and sensor data. Apply the control data on the previous particle set distribution to generate the hypothetical current distribution. For each particle, update the state from X(t-1) to X(t) according to the conditional probability distribution: $p(X(t) \mid X(t-1), U(t))$.

X(t-1) is the state at time t-1.

X(t) is the state at time t.

U(t) is the control data at time t. It includes the motor commands, speed and turn-rate, and elapsed time since last sensor reading.

The exact distribution of X(t) is unknown. We use uniform distribution within certain error range for each particle. This process is easy to implement. And, because we use the mean value of all particles as the estimate of the robot state, the mean of these uniform distributions is close to normal (Gaussian) distribution according to the central limit theorem.

3) Weight calculation

Calculate W, the importance weight of each particle, based on the conditional probability of Z(t), the sensor data reading at time t, assuming the state is X(t).

$$W = p(Z(t) \mid X(t))$$

This conditional probability is difficult to calculate, but it correlates to how well the hypothetic state X(t) matches the sensor data. So, we use a matching score as the weight for each particle.

For laser data, we assume the robot is located at X(t), compute the hypothetical laser readings based on the map generated so far. Then, we count the number of matches with the actual laser readings.

For magnetic field data, we check the match of previously recorded magnetic data on map with the current magnetometer reading.

4) Resample the particle set

Resample the particles with replacement according to their weights. The particles with more weight have higher probability to be re-sampled. It is also called importance sampling.

After this step the particles with higher weight (importance) are sampled more times than other particles. The resulting particle set contains the same number of particles, but more closely represents the actual state distribution.

5) Go to the step 2 for the data at next time point.

In steps 2-4, the particle set for state distribution at (t-1) is updated to the distribution at time (t). After step 4, the global map is updated according to the mean location among the new distribution, and the current laser inputs. Magnetometer data are also recorded on the map.

B. Process the hijack event

A hijack event means the robot is moved to an unknown place without collecting any control or sensor data. To solve the loop closure problem, we need to detect if the robot is visiting a previous place after being hijacked, and determine where the current location on the previously generated map is. This is necessary to keep consistency between map segments generated before and after hijacking.

After hijacking, we reset the particle set to distribute uniformly among the state space, i.e. x, y location is distributed uniformly on the entire map, and yaw angle is distributed uniformly in $[-\pi, +\pi]$. Also, since particles are distributed on the entire space, we increase the number of particles to keep sufficient particle density for finding matches with previous map segment.

To determine the current location relative to the previous map, we run the particle filter steps 2-4 for a number of iterations without updating the map. Then, we estimate the robot location and pose based on the particle set, and resume normal map updating.

The overall flow chart of the algorithm is shown in Fig. 1.

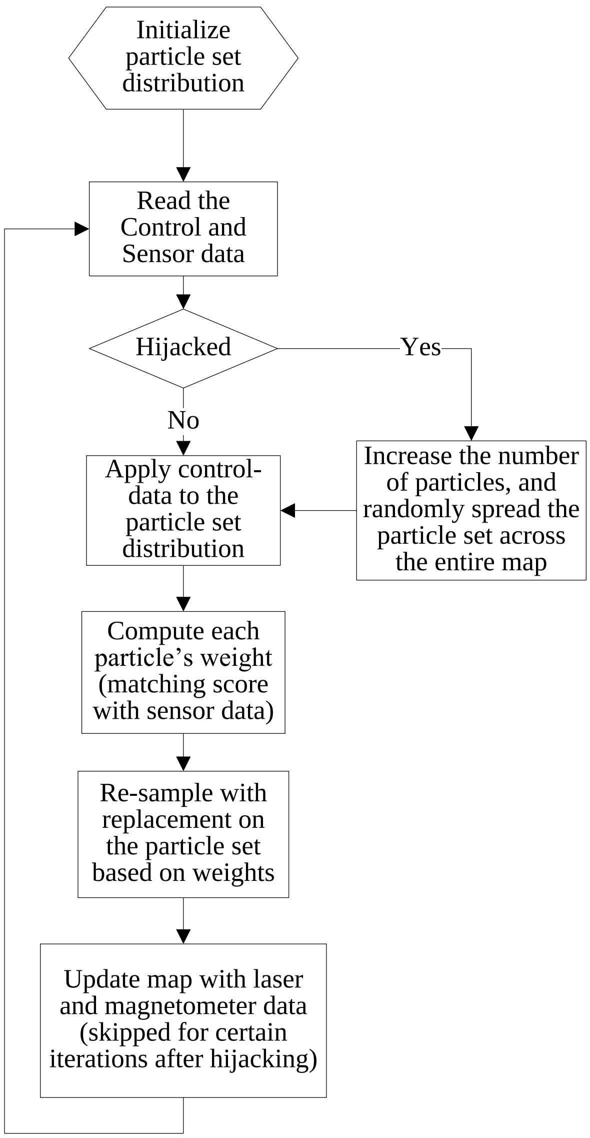


Figure 1. Algorithm flow chart

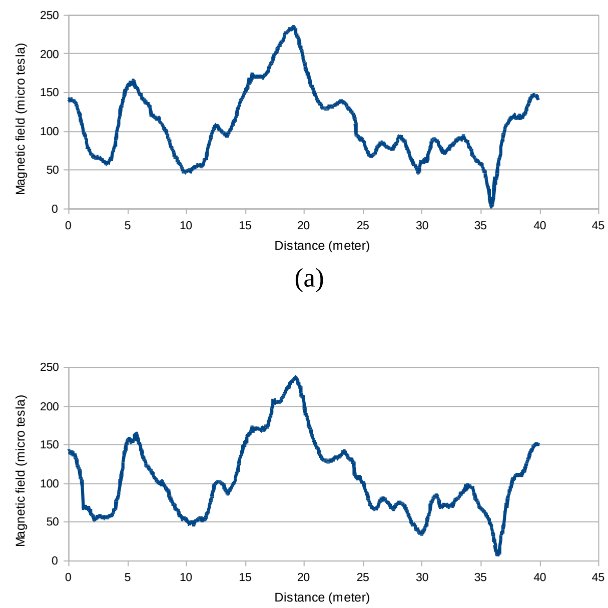
III. EXPERIMENTS AND ANALYSIS

A. Measurements of local magnetic field anomalies

The robot platform is called "Stark", which was designed and built by our IGVC (Intelligent Ground Vehicle Competition) team. It is four wheel driving, skid steering. There are SICK LMS200 laser range finder, PNI MacroMag 3-axis magnetometer, and other sensors on board.

To check the magnetic field anomalies, we collected magnetic field data in two runs. The robot was driven through a hallway in our computer science building. The data from these experiments are shown on Fig. 2. The absolute magnetic field value in the XY plane is used to draw the diagrams. Since the robot is only moving in XY plane, this value is rotationally invariant. The absolute magnetic field value is calculated as SQRT(Bx^2 + By^2). Bx, By are the magnetic field magnitudes in x, y directions.

We have found the magnetic field anomalies are significant among different places, but remain fairly stable between the first run (Fig. 2a), and the second run (Fig. 2b). This indicates it can be used for localization and potentially help solving the loop closure problem of SLAM. We verified this idea in the next set of experiments.



(b)

Figure 2. Magnetic field changes over position, but remains stable across two runs: (a) and (b)

B. Incorporate magnetic field anomaly data into SLAM

SLAM algorithms usually take laser range finder and encoder data. Probabilistic modes can remove most of the random errors in relatively short range. But, after the robot is hijacked to an unknown place, the similarity of laser scan patterns on different part of a building hallway may prevent localizing the robot on the existing map, and impact successful loop closure.

In our approach, magnetic values are overlaid on map grids, missing values are filled in by approximation. We use particle filter algorithm with resampling. The importance weight (W) is calculated based on the matching score between the current sensor readings and data recorded on the existing map. The matching score is to estimate the conditional probability of current laser data (L), current magnetic field value (B), assuming the state is represented by this particle at (x, y, yaw):

$$W = p(L, B \mid x, y, yaw)$$

For comparison, we also run the algorithm without magnetic field anomaly data by simply removing magnetic field value (B) from the equation above:

$$W = p(L \mid x, y, yaw)$$

Fig. 3 shows a set of example control and sensor data. The first line is a set of laser range data with 181 readings. The next line contains the elapsed time since last data point, speed, turn rate, compass direction, and magnetic field value in x,y direction respectively. From these data, we calculate the absolute magnetic field values in the XY plane as the local magnetic anomaly data. The control data, including elapsed time, speed, turn rate, and compass direction, are used by particle filter to generate the hypothetical states. Then, the sensor data, including laser readings and magnetic anomalies, are used for weight calculation and resampling.

```
Laser 181 7.54 7.54 7.54 7.55 7.54 7.48 7.49 7.5 7.53 7.55 7.56 6.9
6.38 6.14 5.73 5.37 5.25 4.95 4.67 2.34 2.33 2.31 2.29 2.27 2.25 2.24
2.22 2.21 2.2 2.19 2.18 2.17 2.16 2.15 2.15 2.13 2.13 2.13 2.13 2.12
2.12 2.12 2.11 2.11 2.11 2.11 2.12 2.12 2.12 2.12 2.12 2.14 2.79 2.75
2.71 2.67 2.65 2.62 2.6 2.56 2.56 2.64 2.74 2.9 3.98 3.97 3.97 3.98 4.01
4.04 4.07 4.14 4.22 4.37 4.65 4.96 5.3 5.72 6.17 6.72 7.37 8.22 9.24
10.54 12.2 81.87 14.7 24.95 81.87 81.87 81.87 81.87 81.87 15.24 14.74
14.69 10.51 9.52 4.77 4.47 81.91 3.52 81.87 81.87 81.87 81.87 81.87
81.87 3.67 3.49 3.33 3.17 3.03 2.91 2.79 2.7 2.61 2.51 2.44 2.36 2.29
2.22 2.16 2.1 2.04 1.9 1.86 1.82 1.77 1.73 1.7 1.69 1.72 1.74 1.77 1.8
1.84 1.86 1.9 1.94 1.97 2.02 2.07 2.11 2.16 2.21 2.26 2.32 2.39 2.46
2.53 2.62 2.57 2.55 2.53 2.51 2.5 2.48 2.47 2.44 2.43 2.41 2.4 2.38 2.38
2.36 2.35 2.34 2.33 2.33 2.31 2.32 2.32 2.3 2.24 2.23 1.86 1.87 1.87
1.87 1.87
CTRMAG 0.080029 0.295245 0 255.34 -0.337986 -1.29203
```

Figure 3. Example of control and sensor data

C. Test loop closure after hijacking

After hijacking, we need to localize the robot on the existing map before start to update the map with new data to keep map consistency. The robot's path in the experiment is shown on Fig. 4. It was driven from the left side of the hallway to the right side (Segment 1), then was hijacked to an unknown place, and driven to the top part of the map. The data were collected into log files during the run. Then, the log files are processed by the SLAM algorithm offline. Our robot doesn't have a hijack sensor, so a "HIJACK" keyword is inserted between the data logs from Segment 1 and 2.

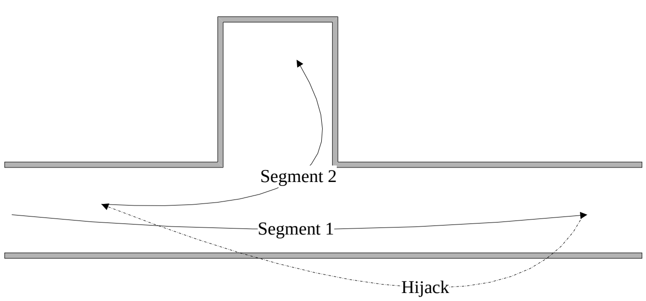


Figure 4. The path traveled by the robot

If we run the SLAM algorithm separately on the data from segment 1 and segment 2, two maps shown on Fig. 5 are generated. Fig. 5a is the segment 1, the trip from left side of the hallway to the right side. The gray area is open space, the dark edges are obstacles, and the white area is unknown space. There are doors, trash cans, and furniture along the hallway. Fig. 5b is the segment 2, which contains part of the same hallway as in segment 1, plus the elevator area shown on the top of the map.

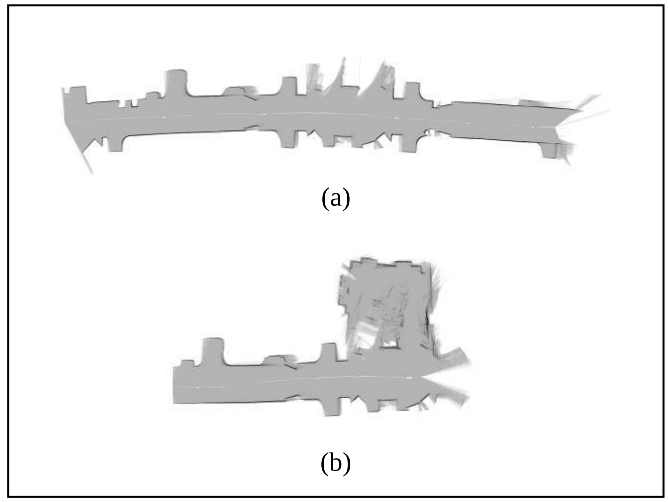


Figure 5. Map segments before and after hijacking. a) Segment 1. b) Segment 2.

For comparison purpose, we run the SLAM program twice on the same data set. At the first time, we use particle filter without considering magnetic anomaly data (Fig. 6 left side). At the second time, we incorporated magnetic anomaly data into the importance weight calculation (Fig. 6 right side).

After hijacking, we don't have any knowledge on the robot location, so reset all particles (red dots, or dark dots on monochrome prints) to spread uniformly on the entire map, as shown on the top two maps on Fig. 6. Then, the particle filter algorithm starts to read new data after hijacking. In each iteration, it reads a new set of control and sensor data, calculate the weights, and resample the particle set. With more iterations, it gradually concentrates the particles to the most likely location.

As shown on the map, there are repetitive patterns on the map due to the arrangements of the doors and furniture in the hallway. With laser data only, it's difficult to distinguish the repetitive patterns. As demonstrated on the maps on Fig. 6 left side, the particles spread across several wide ranges, and failed to converge into the right location.

With both laser and magnetic anomaly data incorporated into the weight calculation, the magnetic field values are overlaid on top of the repetitive patterns. So we have another factor to help us distinguish the patterns. As shown on the maps on Fig. 6 right side, the set of particles converge to the correct location with the help from magnetic anomaly data.

The end results are shown on Fig. 7. The Fig. 7a is the map without considering magnetic field anomaly data, shows the map segment generated after hijacking isn't consistent with the previously generated segment – the same place is shown on two locations on the map. This demonstrates the loop closure problem. The Fig. 7b is the map with magnetic field anomaly data in particle weight calculation, shows the map segment generated after hijacking is consistent with the previously generated segment. It indicates the successful loop closure after hijacking.

IV. Conclusion

This paper presents a method that incorporates measurements of local magnetic field anomalies into SLAM algorithm. We incorporate the magnetic anomaly data into the weight calculation step of the particle filter. By combining laser range finder and magnetometer data, we have one more factor to distinguish visually repetitive laser scan patterns. Also the price of the magnetometer is very affordable.

Our experiments show that the magnetic anomaly based method can correctly localize the robot on the existing map, i.e. close the loop, after being hijacked. On the experiment without considering magnetic anomaly, loop closure has failed. This indicates our method is a low cost and effective way to improve loop closure successful rates, thus enhances the consistency of robotic mapping, which is important for practical robot applications.

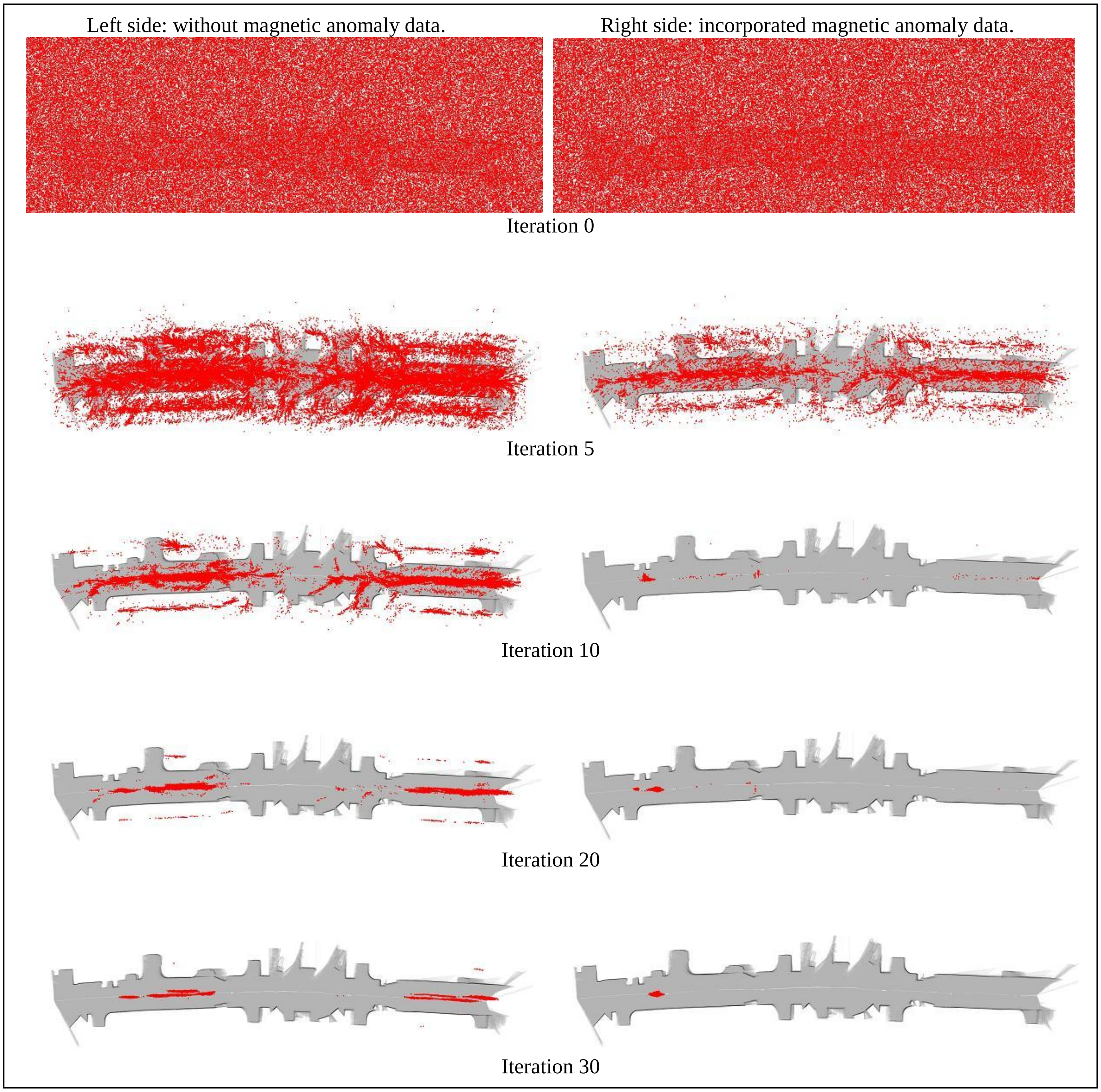
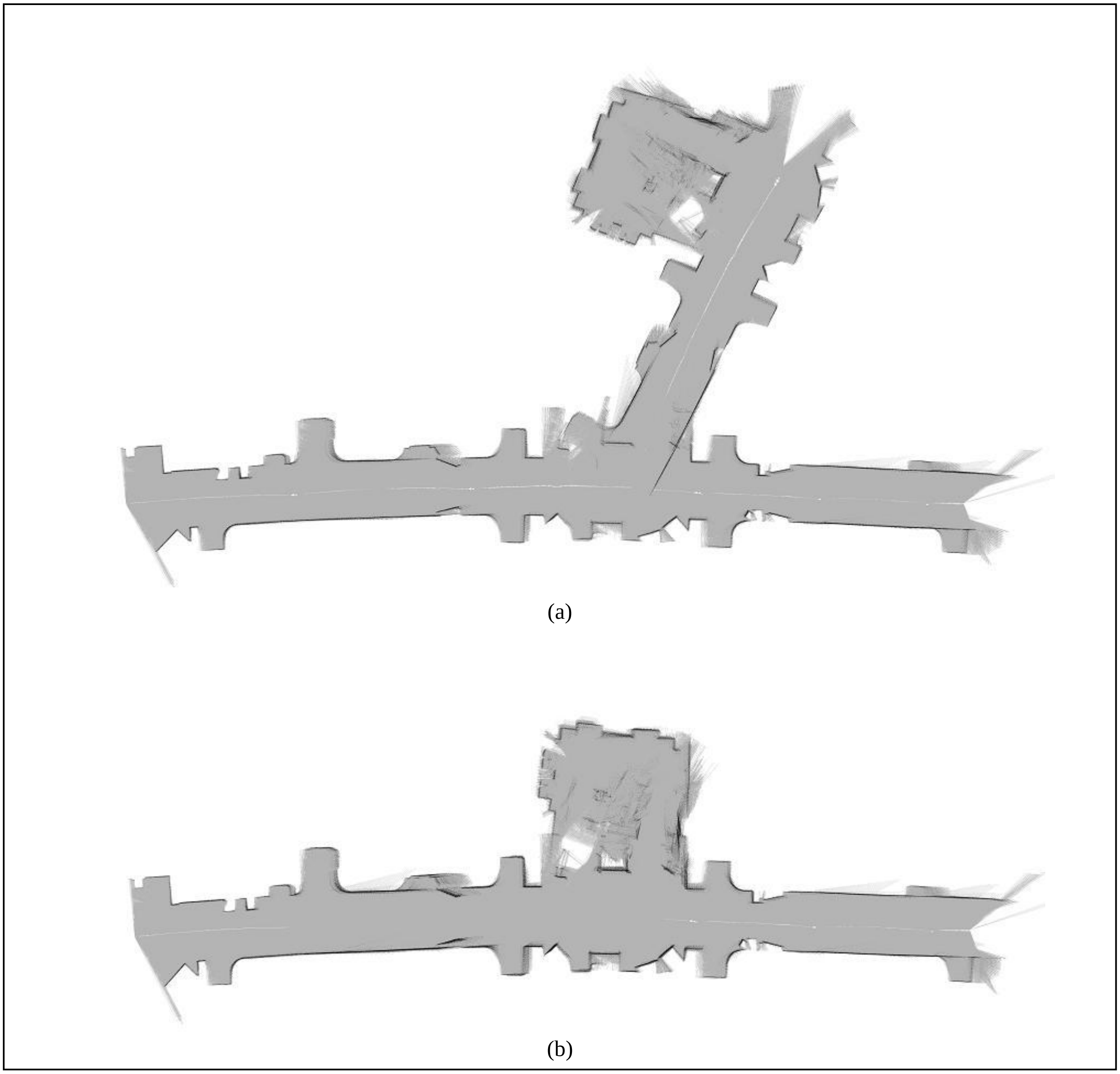


Figure 6. Comparison between particle filter results without vs. with incorporating magnetic anomaly data (left side vs. right side). Initially all particles (the red dots, or dark dots on monochrome prints) spread uniformly on the entire map. The iteration number shown is the number of particle filter iterations after hijacking. The particle filter algorithm gradually concentrates the particles to the most likely location after hijacking. The set of particles converge to the correct location with the help from magnetic anomaly data (maps on the right side), otherwise they distribute through several possible ranges (maps on the left side).



(a)

(b)

Figure 7. a) The final map generated without considering magnetic anomaly data, shows inconsistency of the segments before and after hijacking. b) The final map generated with magnetic anomaly data incorporated, which is a consistent map.

REFERENCES

[1] J.J. Leonard, H.F. Durrant-whyte, "Simultaneous map building and localization for an autonomous mobile robot", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1991), p1442-1447.

[2] S. Thrun, W. Burgard, D. Fox, “Probabilistic Robotics”, MIT Press, 2005.

[3] K. Granstrom, J. Callmer, F. Ramos and J. Nieto, "Learning to Detect Loop Closure from Range Data", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2009), p15-22.

[4] X. Ji, H. Zhang, D. Hai and Z. Zheng, "A Particle-Filter-Based Active Loop Closing Approach to Autonomous Robot Exploration and Mapping", Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA 2008), p824-830.

[5] B. Ferris, D. Fox and N. Lawrence, "WiFi-SLAM Using Gaussian Process Latent Variable Models", Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI 2007), p2480-2485.

[6] J. Haverinen and A. Kemppainen, "A global self-localization technique utilizing local anomalies of the ambient magnetic field", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2009), p3142-3147.

[7] A. Eliazar, R. Parr, “DP-SLAM: Fast, Robust Simultaneous Localization and Mapping without Predetermined Landmarks”, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2003), p1135-1142